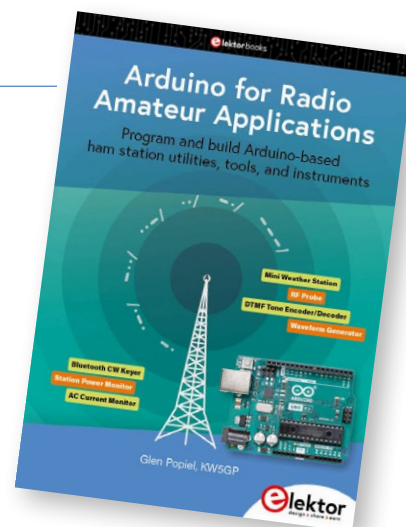


# sonde RF

avec bargraphe à LED



Glen Popiel, KW5GP (ÉTATS-UNIS)

Grâce à cet article, les radioamateurs et les passionnés d'Arduino de tous niveaux pourront concevoir un renifleur de signaux RF à large bande. Ce projet utilise l'amplificateur logarithmique AD8307 d'Analog Devices, un ruban LED RGB WS2812, et un Arduino Nano, accompagné d'un programme spécifique. Glen, KW5GP, c'est à vous de prendre le relais !

**Note de la rédaction.** Cet article est un extrait du livre d'Elektor *Arduino for Radio Amateur Applications*, formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. L'auteur et l'éditeur seront heureux de répondre aux questions – pour les contacter, voir l'encadré **Questions ou commentaires ?**

L'un des atouts de l'Arduino est sa capacité à être utilisé comme un outil pour créer d'autres outils. Si vous êtes comme la plupart des radioamateurs, vous n'avez généralement pas besoin de beaucoup d'équipement de test, à part un multimètre et un wattmètre. Lorsqu'un test nécessite un matériel spécifique que vous ne possédez pas, il est souvent possible d'emprunter ce dont vous avez besoin à un autre radioamateur. La raison principale pour laquelle vous ne possédez pas votre propre arsenal d'équipement de test est que certains peuvent être assez onéreux, ou qu'il ne semble pas justifié d'investir dans des appareils que vous utiliserez très rarement. Il est rare que j'aie besoin d'une sonde RF (figure 1), mais elle peut s'avérer utile lorsqu'il s'agit de vérifier le fonctionnement d'un émetteur, sans disposer d'un wattmètre ou d'un ROS mètre à proximité, ou encore lorsqu'il est néces-

saire de détecter des interférences RF parasites. En effectuant des recherches pour un autre projet, j'ai découvert l'article *RF Driven On-Air Indicator* de Keith Auster Miller, KB9STR, paru dans le numéro d'août 2004 de QST Magazine. Cet article s'inspire lui-même du *No Fibbin' RF Field Strength Meter* de John D. Noakes, VE7NI, publié dans le numéro d'août 2002 du même magazine. Avec quelques modifications mineures, ces projets pourraient être adaptés pour créer une sonde RF qui permettrait à un Arduino de piloter des LED adressables WS2812 remplaçant ainsi un compteur classique pour indiquer l'intensité relative du signal RF.

## Idee de conception et composants principaux

Bien que ce projet puisse paraître relativement simple et direct, je m'attache toujours à réaliser une documentation exhaustive. Disposer de cette documentation est extrêmement bénéfique si jamais je décide de revenir sur ce projet ultérieurement pour le modifier, y intégrer de nouvelles fonctionnalités, etc.

La figure 2 montre le schéma fonctionnel du projet de sonde RF. Outre l'entrée de détection RF, l'Arduino est programmé pour afficher l'intensité du signal sur un ruban à 8 LED RGB WS2812. Cet aspect simplifie grandement ce projet. Au lieu de devoir contrôler des LED RGB individuelles, chacune nécessitant trois broches d'E/S numériques et un total de 24 broches d'E/S pour un affichage à 8 LED, les barres et bandes de LED RGB WS2812 adressables ne requièrent qu'une seule broche d'E/S numérique. Il est également possible d'utiliser une puce de contrôle de bande de LED tel que le MAX7219 d'Analog Devices qui n'utilise qu'une seule broche d'E/S numérique. Mais pourquoi introduire la complexité d'une puce à 24 broches quand elle n'est pas nécessaire ?

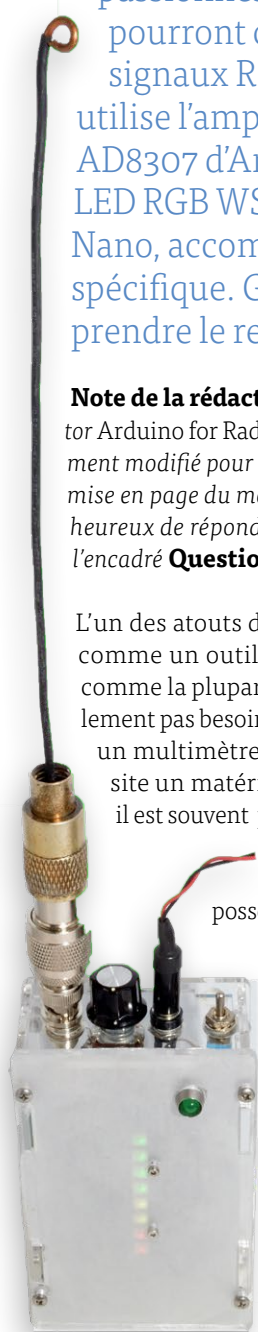


Figure 1. La sonde RF terminée.

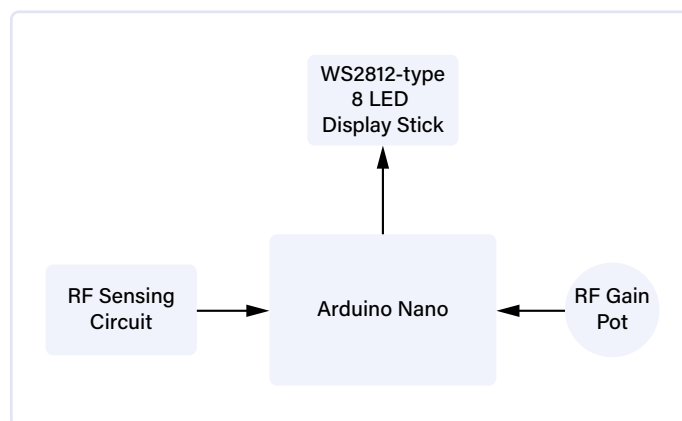


Figure 2. Schéma fonctionnel de la sonde RF.

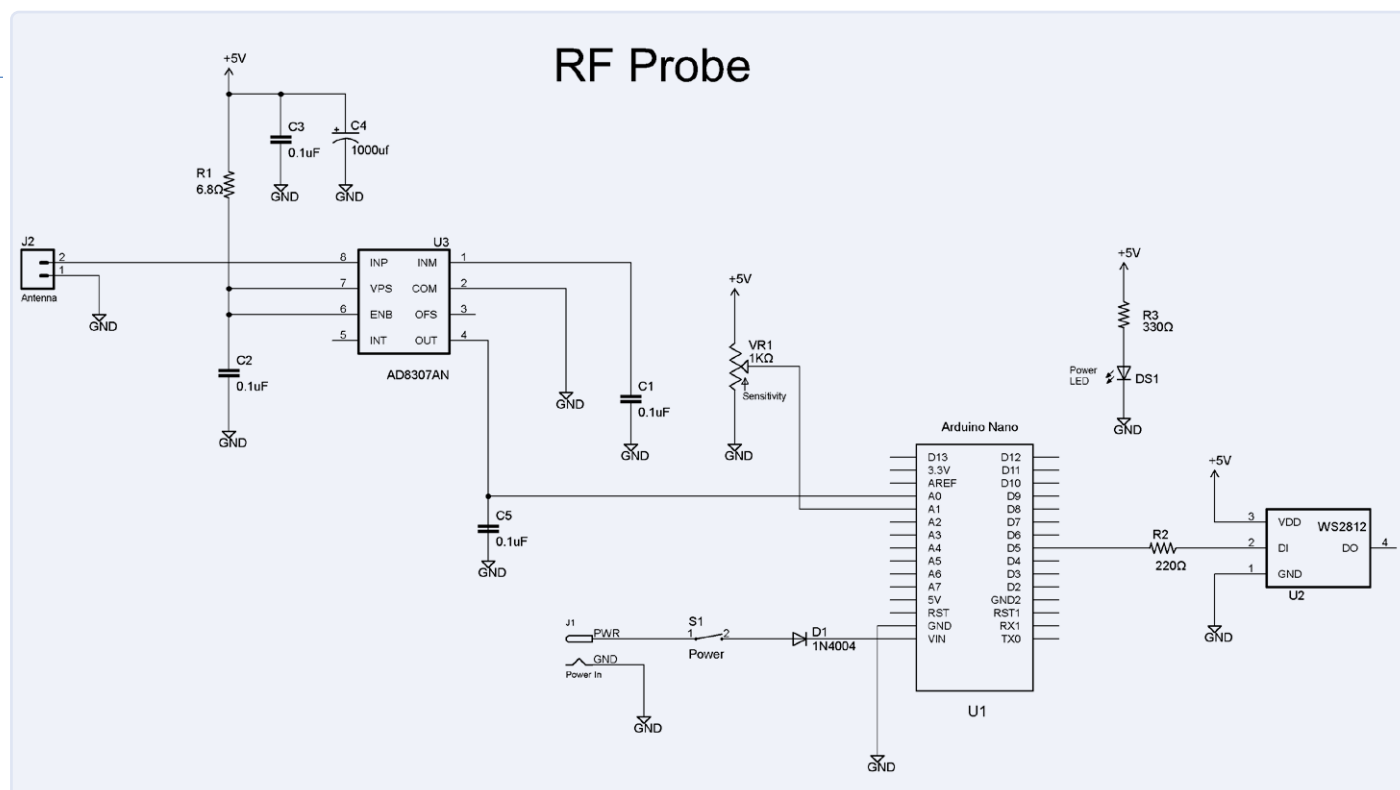


Figure 3. Schéma de la sonde RF.

## Schéma et construction

La construction de la sonde RF diffère légèrement de l'approche habituelle décrite dans le reste du livre. Au lieu de prototyper l'unité de détection RF sur une plaque d'essai et de se préoccuper des interférences RF dues au câblage, le circuit de détection RF est assemblé et monté sur la même plaque que l'Arduino Nano. Plutôt que d'utiliser le circuit d'ampli-op présenté dans l'article sur l'indicateur *on-air piloté par RF*, j'ai décidé de moderniser un peu le circuit en utilisant un amplificateur logarithmique AD8307 d'Analog Devices, comme illustré dans le schéma de la sonde RF à la **figure 3**. Capable de détecter des signaux RF jusqu'à 500 MHz, la sonde RF est ainsi adaptée à presque tous vos besoins de détection hors des bandes de fréquences des téléphones cellulaires et du Wifi. Puisque la sortie de l'AD8307 est une tension analogique représentant l'intensité du signal, nous pouvons connecter la sortie de l'AD8307 directement à l'entrée analogique A0 de l'Arduino Nano.

Le connecteur d'antenne est fixé sur un panneau latéral d'un boîtier Solarbotics et se connecte à la carte de circuit imprimé de la sonde RF en utilisant un connecteur mâle à pas de 0,1 pouce (2,54 mm) et un connecteur femelle DuPont. Cette configuration permet de retirer facilement la carte pour rectifier toute erreur de câblage qui pourrait survenir. La puce de l'amplificateur logarithmique AD8307 est montée dans un socket. Cette précaution est utile pour les occasions où elle pourrait être accidentellement exposée à une puissance de 100 watts, au risque de la faire surchauffer et de laisser échapper de la fumée.

La sonde RF est construite sur une carte de prototypage de 60 mm × 80 mm (**figure 4**) et est installée dans un boîtier Solarbotics Mega SAFE transparent. Un connecteur de châssis coaxial SO-239 a été monté sur le boîtier, ce qui vous permet d'utiliser différentes antennes pour la capture des signaux RF. Nous avons soudé un segment de 10 pouces (25 cm) de fil solide #14 AWG au conducteur central d'un connecteur coaxial PL-259 pour faire office d'antenne pour circuit de détection RF.

## Entrée, sortie et commandes

Pour ce projet, nous utiliserons le convertisseur analogique-numérique (ADC) 10 bits intégré de l'Arduino Nano pour convertir la sortie de tension analogique de l'AD8307 en une valeur numérique. Cette valeur sera ensuite convertie pour déterminer le nombre de LED à allumer sur une bande de 8 LED RGB WS2812, indiquant ainsi l'intensité du signal. Pour la sonde RF, l'utilisation d'un écran TFT pour l'affichage n'est pas nécessaire car nous recherchons uniquement une indication relative de l'intensité du champ RF, ce à quoi se prêtent idéalement les bandes de LED WS2812. Nous utiliserons un potentiomètre de 1 kΩ pour régler la sensibilité RF de la sonde.

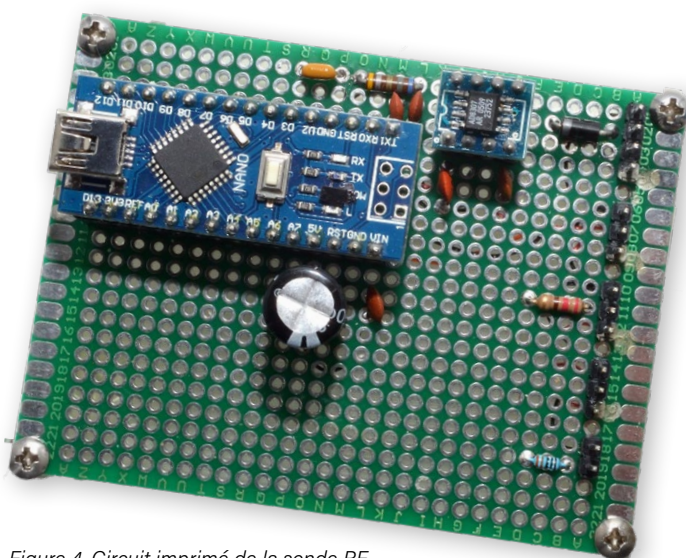


Figure 4. Circuit imprimé de la sonde RF.

## Croquis Arduino

Le programme pour la sonde RF est en fait assez simple. Grâce à l'organigramme du projet illustré dans la **figure 5**, le code est assez concis, ce qui démontre l'efficacité de l'Arduino pour gérer les tâches d'entrée/sortie simples requises dans ce projet. Le programme se contente de lire la tension analogique de l'AD8307 et de sortir une représentation graphique de cette tension d'entrée sur la bande de LED.

Le croquis complet ainsi que les bibliothèques de ce projet sont disponibles en ligne [1]. Le croquis utilisera quatre LED vertes, deux jaunes et deux rouges, pour indiquer l'intensité du signal détecté, comme le montre la **figure 6**. Vous remarquerez que j'ai utilisé un petit morceau de papier de soie collé à l'intérieur du couvercle transparent du boîtier comme diffuseur. Vous pourriez également utiliser l'un des boîtiers translucides de Solarbotics.

Concernant le code, nous commençons par inclure incluant la bibliothèque *Adafruit\_NeoPixel* pour piloter le ruban de LED RGB adressable. Il existe de nombreuses bibliothèques pour les LED RGB adressables de type WS2812, mais je préfère toutes les fonctionnalités et la flexibilité de la bibliothèque Adafruit.

```
#include <Adafruit_NeoPixel.h>
```

Ensuite, nous allons définir le nombre de LED sur le ruban de LED adressable et la broche d'E/S à laquelle il sera connecté. L'un des avantages du WS2812 est sa capacité à gérer en interne la modulation de largeur d'impulsion (PWM) nécessaire pour contrôler l'intensité de chaque LED, ce qui permet d'utiliser n'importe quelle broche d'E/S numérique de l'Arduino pour la connexion à la ligne de données. Les rubans et bandes de LED RGB adressables WS2812 ne nécessitent qu'une seule broche d'E/S numérique pour fonctionner. Nous utiliserons également les directives *#define* pour simplifier la configuration des couleurs des LED.

```
#define NUM_LEDS 8
#define LED_PIN 5
#define RF_input_pin A0
#define gain_pin A1
#define led_Red 255, 0, 0
#define led_Green 0, 255, 0
#define led_Yellow 255, 255, 0
#define led_brightness 50
#define update_delay 200
```

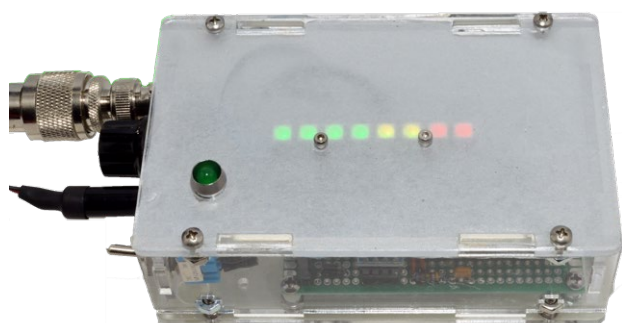


Figure 6. Le boîtier de la sonde RF.

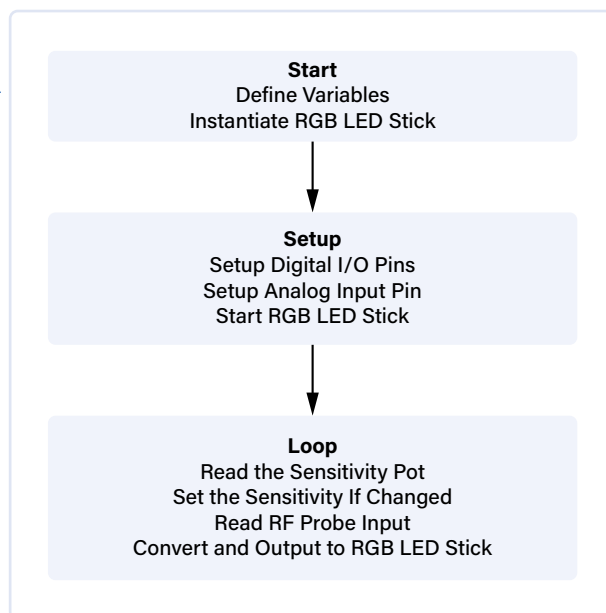


Figure 5. Organigramme de la sonde RF.

Après avoir déclaré les variables nécessaires à ce croquis, nous procéderons à l'instanciation du ruban de LED.

```
int signal_strength = 0;
int effective_strength = 0;
int max_strength = 1000;
int LEDsToLight = 0;
int rf_gain;
```

```
Adafruit_NeoPixel pixels
(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
```

La fonction *setup()* de la sonde RF est conçue pour être aussi concise et efficace que possible. Elle se limite à initialiser le ruban de LED RGB adressable, à éteindre toutes les LED, puis à invoquer la méthode *show()* de la classe *NeoPixel* pour activer l'effacement des LED.

```
pixels.begin(); // INITIALIZE NeoPixel strip object

pixels.clear(); // Set all pixel colors to 'off'
pixels.show();
```

Dans la boucle *loop()*, nous commençons par lire la tension analogique du potentiomètre de gain RF connecté à la broche d'entrée analogique 1 de l'Arduino. Ensuite, nous mesurons l'intensité du signal RF sur la broche d'entrée analogique 0. Nous utilisons la valeur obtenue du réglage de gain pour ajuster la fonction *map()* de l'Arduino, ce qui permet de régler la sensibilité des LED de la sonde RF.

Il s'agit d'un moment idéal pour introduire la fonction *map()* d'Arduino, utilisée pour modifier l'échelle de vos données. Le signal RF détecté, dont l'amplitude varie de 0 à 5 V et est mesurée par l'AD8307, est converti d'une plage numérique possible de 0 à 1023 (correspondant à la résolution de 10 bits de l'Arduino) à une échelle de 0 à 8, qui représente le nombre de LED sur le ruban. La valeur obtenue nous indique ainsi le nombre de LED à allumer pour indiquer l'intensité du champ RF

```
LEDsToLight = map(effective_strength, 0, max_strength, 0, NUM_LEDS);
```

```
if (LEDsToLight < 0)
{
    LEDsToLight = 0;
}
```

```
if (LEDsToLight > NUM_LEDS)
{
    LEDsToLight = NUM_LEDS;
}
```

Maintenant que nous avons déterminé le nombre de LED à allumer, il nous faut choisir la couleur des LED allumées. Comme convenu, les quatre premières LED s'illumineront en vert, les deux suivantes en jaune, et les deux dernières en rouge. Nous commençons par effacer la configuration précédente des LED, puis nous activons les LED correspondant à l'intensité actuelle du signal RF.

```
pixels.clear();
```

Ensuite, les couleurs et le nombre souhaité de LED à allumer sont définis.

```
pixels.setBrightness(led_brightness);
```

```
for (int led = 0; led < LEDsToLight; led = led + 1)
{
    if (led <= 3)
    {
        pixels.setPixelColor(led, pixels.Color(led_Green));
    }
    if (led > 3 && led <= 6)
    {
        pixels.setPixelColor(led, pixels.Color(led_Yellow));
    }
    if (led > 5)
    {
        pixels.setPixelColor(led, pixels.Color(led_Red));
    }
}
```

Enfin, nous utilisons la fonction `pixels.show()` pour allumer le nombre désiré de LED dans les couleurs spécifiées.

```
pixels.show();
```

## LIEN

[1] Croquis et bibliothèques : <https://tinyurl.com/kw5gpelektor>

## Liste des composants

### Résistances

R1 = Résistance de 6,8 Ω, 1/8 W

R2 = Résistance de 220 Ω, 1/8 W

R3 = Résistance de 330 Ω, 1/8 W

VR1 = Potentiomètre de 1 kΩ

### Condensateurs

C1, C2, C3, C5 = 0,1 µF, 25 V condensateur

C4 = 1000 µF, condensateur de 25 V

### Semi-conducteurs

D1 = diode 1N4004

DS1 = LED de 5 mm

U2 = bâton WS2812 à 8 LEDs

U3 = AD8307 CI amplificateur logarithmique

### Divers

J1 = Prise d'alimentation DC

S1 = Interrupteur SPST

U1 = Arduino Nano

Boîtier : Solarbotics Mega S.A.F.E.

La valeur `update_delay` est utilisée pour contrôler l'intervalle entre les itérations du croquis. Je l'ai réglée à 200 ms dans le programme, ce qui permet d'obtenir cinq mises à jour par seconde. Vous pouvez ajuster cette valeur selon vos besoins. Une valeur de 0 mettra à jour les LED sur le ruban aussi rapidement que l'Arduino le permet, offrant ainsi un affichage presque en temps réel de l'intensité du signal. Il est également possible d'utiliser cette technique pour créer un indicateur de niveau audio dynamique pour les applications audio.

```
delay(update_delay);
```

## Idées d'amélioration

Ce projet se prêterait parfaitement à une réalisation dans une boîte de bonbons Altoids, en utilisant un Arduino Nano. En optant pour un connecteur SMA plutôt qu'un SO-239, il serait possible de tout assembler à l'intérieur de la boîte, avec juste une petite ouverture pour le ruban de LED adressables. Une autre modification envisageable serait de remplacer le ruban de LED par un écran OLED compact, ce qui permettrait à la sonde RF d'afficher une représentation numérique de l'intensité relative du champ RF. ◀

240326-04

## Questions ou commentaires ?

Envoyez un courriel à l'auteur ([glenpopiel@gmail.com](mailto:glenpopiel@gmail.com)), ou contactez Elektor ([redaction@elektor.fr](mailto:redaction@elektor.fr)).



## Produits

> Glen Popiel, *Arduino for Radio Amateur Applications* (Elektor 2024)

Version papier : [www.elektor.fr/20814](http://www.elektor.fr/20814)

version numérique : [www.elektor.fr/20815](http://www.elektor.fr/20815)